Image Defocus Analysis for Finger Detection on A Virtual Keyboard

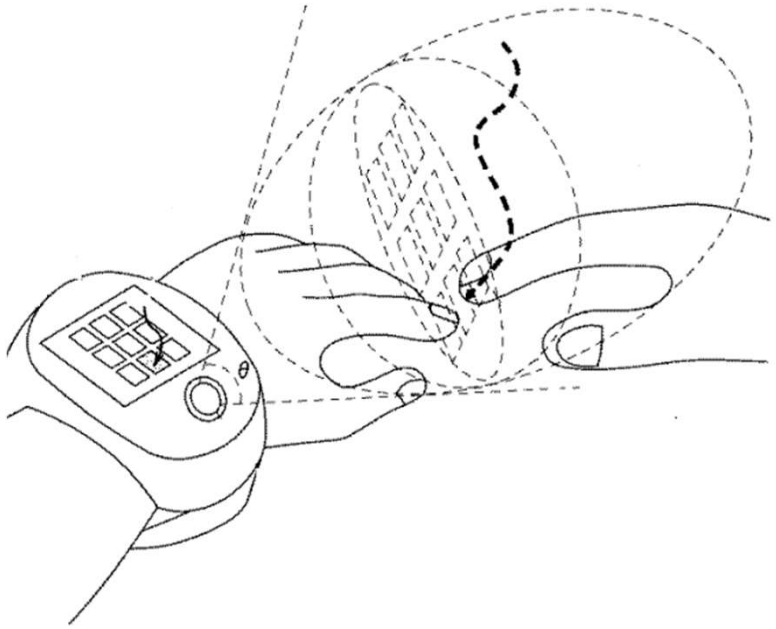
Michio Miwa
Member
Urayasu city Chiba Japan
miwa.michio@gmail.com

Kenji Honda
Tokyo University of Marine Science and Technology
Faculty of Marine Technology
khonda@kaiyodai.ac.jp

Makoto Sato
Professor Emeritus
Tokyo Institute of Technology
mkt.sato@gmail.com

1. The target of our research

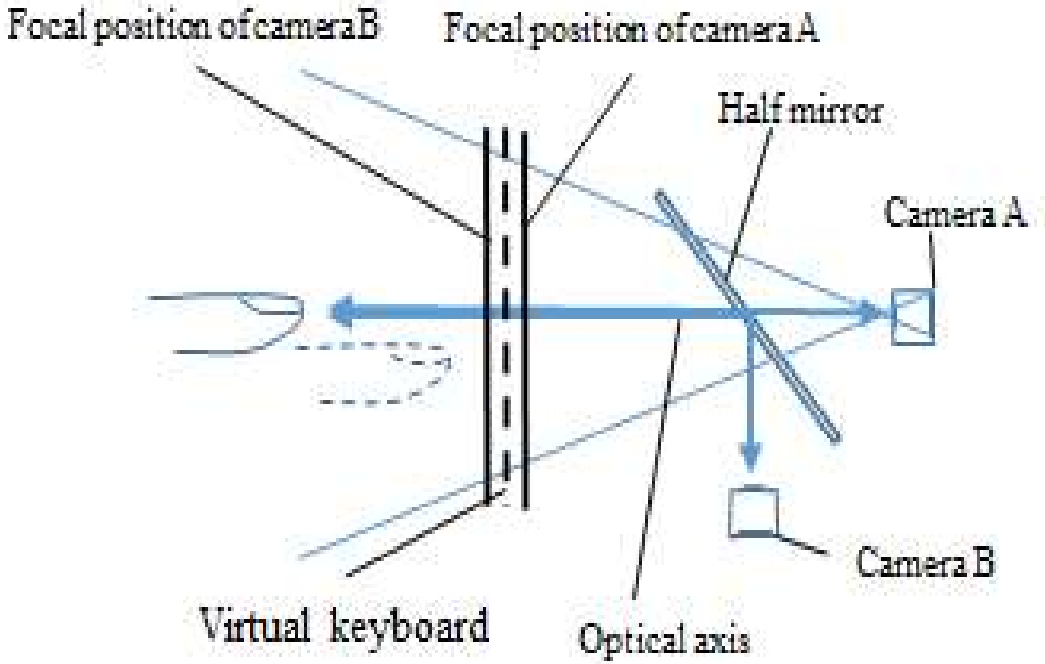
The target of our research is the virtual keyboard of a smart watch.
The touching area of the smart watch is very small.
We can overcome this problem by using the virtual keyboard.



2. The architecture of the virtual keyboard

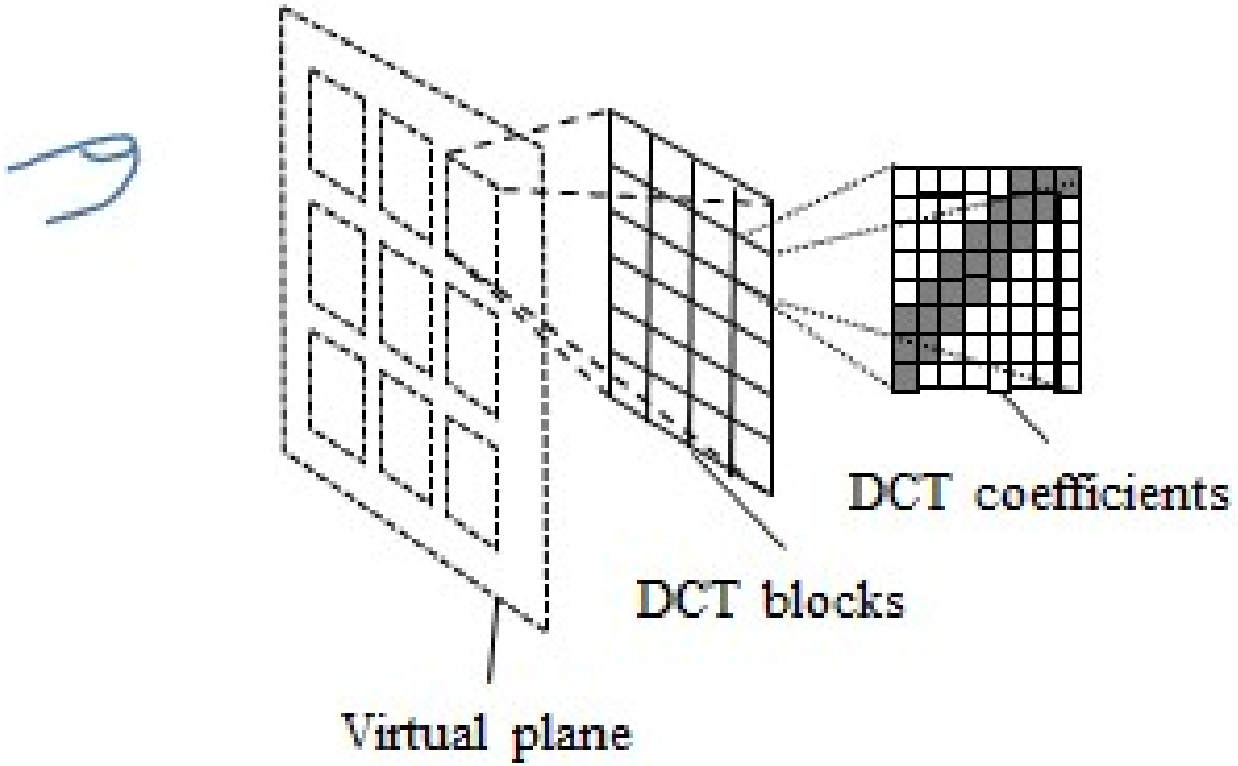
Our virtual keyboard uses 2 cameras and a half mirror. The light goes direct through the half mirror is caught by camera A. The light reflected by the half mirror is caught by camera B.

The focal distance of the camera A and camera B are different with each other a little. The virtual keyboard is set at the middle point virtually



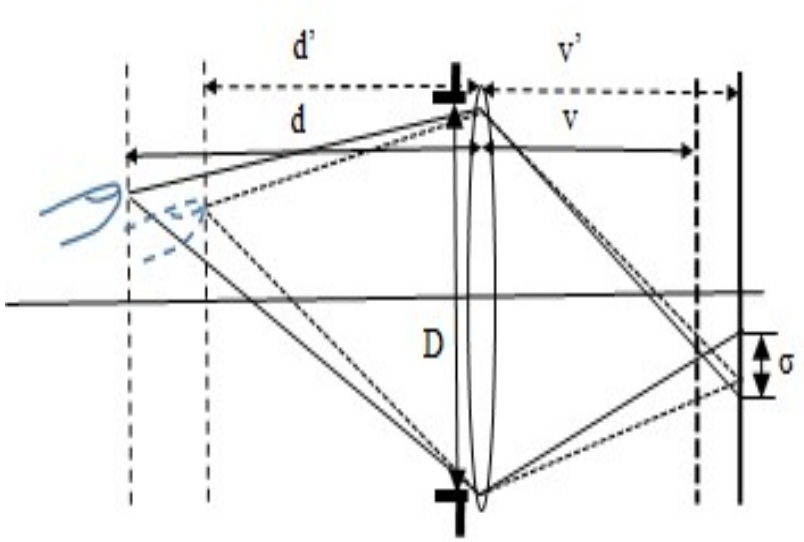
3. Detecting blur information

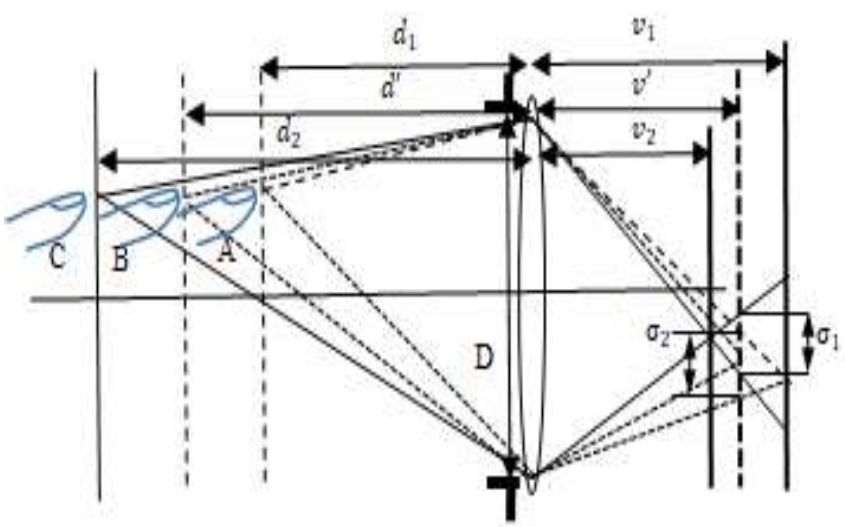
We detect blur information by using DCT coefficients.
The key area on the captured image is divided to 8 * 8 pixels blocks.



4. The depth resolution of the virtual keyboard

The length d1 and the length d2 is near position and far position where we can distinguish if the finger is out of focus or not when the camera is focused at d'.





$$d_1 = \frac{d'f(D+\sigma_1)}{d'\sigma_1+Df} \qquad \text{eq.1}$$

$$d_2 = \frac{d'f(\sigma_2-D)}{d'\sigma_2-Df} \qquad \text{eq.2}$$

D: radius of the lens

f: focal distance of the lens

5. The specification of our camera

Table 1 is the specification of our the camera. The original pixel size of the CMOS is 2592 * 1944. After taking the image from the camera, we shrink the image to 320 * 240 for reducing calculation time. Then the resolution of the image is estimated as 0.017813 mm per pixel.
The minimum radius of the blur circle we can distinguish is the resolution of the camera (CCD resolution). The table 2 shows d1 and d2 when σ is 0.017813 (our camera).
We use raspberry-pie computer and OPENCV library for computing the DCT coefficient.

camera	CMOS size	5.7mm × 4.3mm
	pixel size	320 × 240 (2592 × 1944)
	ρ	0.017813 (0.002199)
lens	D	10
	f	16
cpu	raspberrypie	ARM 1.4Ghz

Table 1

d'	200	250	300
d 1	195.9924	243.6637	290.8214
d 2	204.1902	256.6991	309.8129

(σ = 0.017813)

Table 2

6. The sum of DCT coeffients

The DCT sum is dependent on the filter type

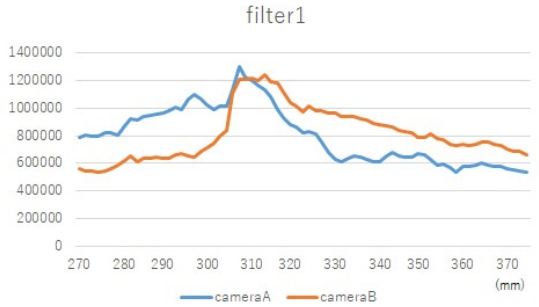
Filter 1 : Low frequency
Filter 2: Middle frequency
Filter 3: High frequency

DCT sum : The value of umming up the DCT coefficient where the value is 1

Cross point is the position where the DCT sum of the both camera is same.

The sum of middle frequency is most effective for detecting the finger touch

```
[[0.1.1.1.1.0.0.0]
 [1.1.1.1.0.0.0.0]
 [1.1.1.0.0.0.0.0]
 [1.1.0.0.0.0.0.0]
 [1.0.0.0.0.0.0.0]
 [0.0.0.0.0.0.0.0]
 [0.0.0.0.0.0.0.0]
 [0.0.0.0.0.0.0.0]
```




```
[[0.0.0.0.0.0.1.1].
 [0.0.0.0.0.1.1.1].
 [0.0.0.0.1.1.1.1].
 [0.0.0.1.1.1.1.0].
 [0.0.1.1.1.1.0.0].
 [0.1.1.1.1.0.0.0].
 [1.1.1.1.0.0.0.0].
 [1.1.1.0.0.0.0.0]]
```

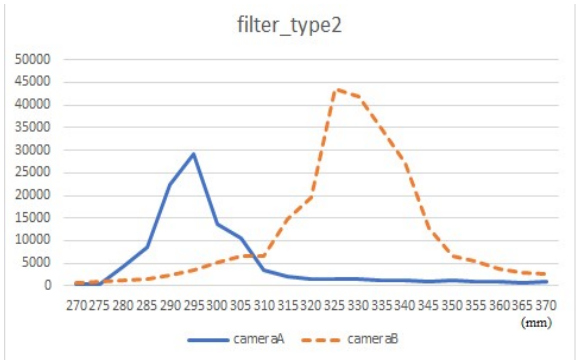
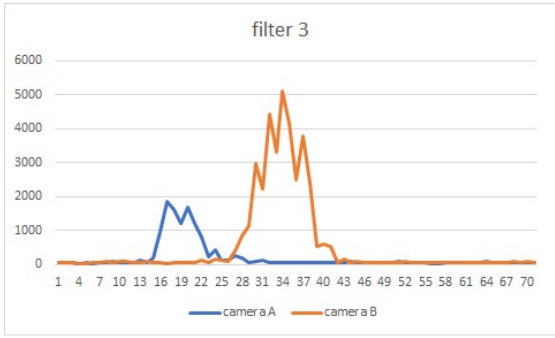


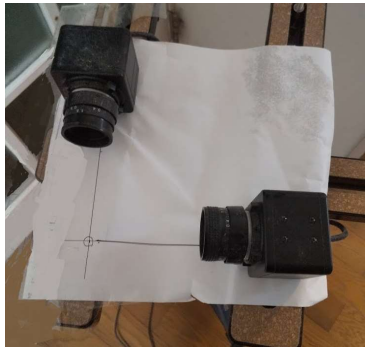
Fig. 6

```
[[0.0.0.0.0.0.0.0].
 [0.0.0.0.0.0.0.0].
 [0.0.0.0.0.0.0.0].
 [0.0.0.0.0.0.0.1].
 [0.0.0.0.0.0.1.1].
 [0.0.0.0.0.1.1.1].
 [0.0.0.0.1.1.1.1].
 [0.0.0.1.1.1.1.1]]
```



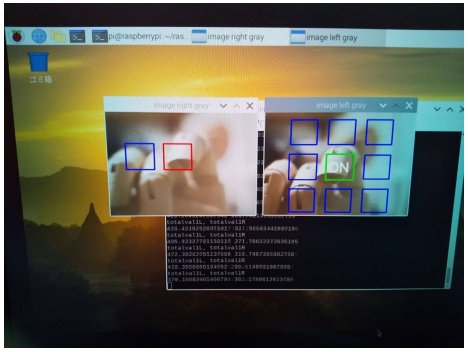
7. Arrangement of the camera

2 USB cameras are arranged
whose focus axes are perpendicular with each other



Half mirror (in the box) is set

Virtual key is shown on the window
Right is the picture of the camera A
Left is the picture of the camera B
(Currently two keys (center and middle row left column key are checked)



When the finger approached, the middle key is pushed